C++ Code Structure and Algorithm Simulator

Ben L. Saminiano
Bicol University Polangui Campus
Polangui, Albay, Philippines

ABSTRACT

The study was mainly intended to develop a C++ Code Structure and Algorithm Simulator that will be used as a supplementary material in teaching computer programming in C++ language, being the vital programming language used in the industry and academe today. Specifically, it answered the following questions: (1) What C++ Code Structure and Algorithm Simulator can be developed? (2) What is the perception of the respondents on the functionality of the simulator? (3) What is the level of acceptability of the simulator? and (4) Is there significant difference in the level of acceptability of the simulator between the students and instructors?

The developmental research was used by the researcher in developing the simulator guided by the phases of Rapid Application Development. The Adobe Flash CS3, ActionScript and Adobe Photoshop were utilized in the construction of the system. The developed simulator contains discussions of the concepts of C++, sample programs and simulated problems. These inclusions were based from the course syllabus of the subject Introduction to Computer Science and guided by the CHED minimum requirements.

After the study was developed, the researcher used the descriptive method to give descriptions and interpretations on how the simulator was developed and how the results of the statistical methods were applied.

The results showed that the simulator was functional and strongly accepted by the students and instructors. Further, there is no significant difference between the acceptability of the students and instructors. Hence, the simulator, and the content and programming problems were comprehensible on the level of both students and instructors.

Keywords: C++, Simulator, Computer-Aided Instruction, Programming

1. INTRODUCTION

Computer-Aided Instruction (CAI), is a diverse and rapidly expanding spectrum of computer technologies that assist the teaching and learning process. CAI is also known as computer-assisted instruction. Examples of CAI applications include guided drills and practice exercises, computer visualization of complex objects, and computer-facilitated communication between students and teachers.

Affordable quality education offered by Bicol University (2012) as cited in its vision, “A University of Excellence characterized by scholarship engagement for the community towards sustainable development”; and mission which is “to give professional and technical training and provide advanced and specialized instruction in literature, philosophy, the sciences and arts, besides providing for the promotion of scientific and technological researches. Hence, BU graduates shall be distinguished by the four pillars of the University: Leadership, Scholarship, Character and Service” contributes much to the growing demand on higher education. This is evident in the increase in enrollees especially in technical courses like engineering and information technology education. Bicol University Polangui Campus (BUPC) as one of the campuses of the university aims to produce graduates that effectively and efficiently meet the scientific, technological needs of business, industries and communities in the local and global economy.

The researcher as a computer engineer and an instructor of BUPC conducted this research to have a supplemental instructional material in teaching computer programming subject in C++, being the vital programming language used in the industry and academe today. Also, it provides an efficient and effective teaching-learning method both to the faculty and the students. Further, it changed the impression of the students on C++ from a difficult to an interesting subject.

2. OBJECTIVES

The goal of the study was to develop a C++ Code Structure and Algorithm Simulator. Specifically, it answers the following questions:

1. What C++ Code Structure and Algorithm Simulator can be developed?
2. What is the perception of the respondents on the functionality of the simulator?
3. What is the level of acceptability of the simulator?
4. Is there significant difference in the level of acceptability of the simulator between the students and instructors?

3. MATERIALS AND METHODS

3.1 Research Method

Developmental research is the systematic study of designing, developing and evaluating instructional programs, processes and products that must meet the criteria of internal consistency and effectiveness (Sherry...
The descriptive research is an attempt to examine situations in order to establish what is normal – what can be predicted to happen again under the same circumstances and relies on observation as means of collecting data (Clarke, 2005).

The developmental research was used by the researcher in developing the simulator. After the study was developed, the researcher used the descriptive method to give descriptions and interpretations on how the simulator was developed and how the results of the statistical methods were applied.

3.2 Respondents of the Study

The respondents of the study were composed of thirty first year Information Technology Education students who took the subject Introduction to Computer Science during the first semester, academic year 2012 – 2013 of BUPC; and five instructors from BUPC and Ateneo de Naga University. They were selected using the nonprobability sampling, purposive sampling. Both students and instructors answered the questionnaires for acceptability of the simulator. Further, the instructors were given additional questionnaire for the test of functionality.

3.3 Data Gathering Tools

In conducting the research, the instrument used was documentary analysis and questionnaire to gather necessary data.

3.3.1 Documentary Analysis

The researcher as an instructor of the subject Introduction to Computer Science for three years, analysed the record of the students. Also, observation and interview were done to further interpret the data.

3.3.2 Questionnaire

The researcher used questionnaires for this study as the instrument to gather data. The questionnaires were utilized to test the functionality and acceptability of the simulator. The test for functionality was determined by two criteria, either functional or non-functional. For the test of acceptability, the evaluation of the simulator and was divided into three categories: simulator, content and objectives. The rating scale was set to 5 for “Strongly Acceptable,” 4 for “Acceptable,” 3 for “Moderately Acceptable,” 2 for “Fairly Acceptable” and 1 for “Not Acceptable.” The responses were treated in such a way that those with highest score rating indicated to be most favourable.

Preparation of Questionnaire

The ideas in formulating the questionnaire was gathered by scrutinizing the designs and layouts of the questions utilized by the researcher who conducted this study.

Validation of Questionnaire

The questionnaires were drafted by the researcher with the help of some knowledgeable persons on the field of research. Then it was submitted to the researcher’s adviser for corrections, suggestions and comments. The comments and suggestions were applied and the final copies were reproduced for distribution to the respondents.

Administration of the Questionnaire

Upon the approval of the proposal, the permission to conduct the study was obtained by the researcher from the Deans of Bicol University Polangui Campus and Ateneo de Naga University. A letter of request duly signed by the researcher and noted by the Dean of College of Engineering of CSPC was given to the Deans of concerned institutions.

3.4 Research Procedure

The following procedures were followed by the researcher in the development of the instructional material applying the Rapid Application Development (RAD) by Kendal and Kendal (2011) which is a complete methodology with a four stage life cycle.

3.4.1 Requirements Planning

The researcher reviewed the course description and course syllabus of the subject based on the minimum requirements set by the CHED. He also gathered the necessary data that were used in the simulator.

3.4.2 User Design

During this phase, the researcher designed the user interface of the system based on the analysed gathered data. The necessary inputs, processes and outputs of the simulator were identified.

3.4.3 Construction

At this point, construction of the system was done. The researcher used the Adobe Flash CS3 and ActionScript as the programming tools.

3.4.5 Cutover

The finished simulator undergone testing for functionality before it was presented to the faculty and students for evaluation. The questionnaires were prepared, validated and distributed to the respondents to evaluate the level of acceptability of the simulator. The results of the
evaluation were analysed and the significant difference in the level of acceptability of the simulator between the students and instructor was tested.

3.5 Statistical Tools Used

The statistical tools used in this study were simple arithmetic mean and t-test.

3.5.1 Arithmetic Mean

The varying degree of the points of the respondents will be 5 for “Strongly Acceptable,” 4 for “Acceptable,” 3 for “Moderately Acceptable,” 2 for “Fairly Acceptable,” and 1 for “Not Acceptable.” The sum will be divided by ‘n’ to find the arithmetic mean. The formula for the arithmetic mean (Coronel, et al., 2004) is

\[ \mu = \frac{\sum x}{n} \]

where \( \mu \) stands for the arithmetic mean which corresponds to the average scale value rated to all the items in each category. \( \sum x \) stands for the summation of the rate given by the respondents to a particular item, and \( n \) for the number of the total respondents.

Table 1 shows the scale, range of values and interpretations as applied to evaluate along its acceptability the C++ Code Structure and Algorithm Simulator.

<table>
<thead>
<tr>
<th>Scale</th>
<th>Range</th>
<th>Verbal Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>4.21 – 5.00</td>
<td>Strongly Acceptable</td>
</tr>
<tr>
<td>4</td>
<td>3.41 – 4.20</td>
<td>Acceptable</td>
</tr>
<tr>
<td>3</td>
<td>2.61 – 3.40</td>
<td>Moderately Acceptable</td>
</tr>
<tr>
<td>2</td>
<td>1.81 – 2.60</td>
<td>Fairly Acceptable</td>
</tr>
<tr>
<td>1</td>
<td>1.00 – 1.80</td>
<td>Not Acceptable</td>
</tr>
</tbody>
</table>

3.5.2 The t-test.

The t-test was used to determine the significant difference between the level of acceptability of the students and instructors of the C++ Code Structure and Algorithm Simulator. The formula for t-test according to Broto (2006) for two independent samples/groups is

\[ t = \frac{x_1 - x_2}{\sqrt{\frac{SS_1 + SS_2}{n_1 + n_2 - 2}} \left( \frac{1}{n_1} + \frac{1}{n_2} \right)} \]

where \( t \) stands for the t-test, \( x_1 \) for the mean of the instructors, \( x_2 \) for the mean of the students, \( SS_1 \) for the sum of the squares of the instructors, \( SS_2 \) for the sum of the squares of the students, \( n_1 \) for the number of instructors, and \( n_2 \) for the number of students.

4. RESULTS AND DISCUSSIONS

4.1 Development of the Simulator

The simulator was guided by the paradigm of software development. After determining the purpose and specifications of the software, the researcher developed a plan for a solution.

The researcher started with the creation images for the system interface using Adobe Photoshop. Next, he started designing the main form of the simulator using Adobe Flash Professional CS3 for the Graphical User Interface for Windows platform. The programming used was ActionScript to make necessary animations of the simulator.

The researcher performed self-test to eliminate bugs or errors. This was done by testing the different features of the simulator. Each feature was repeatedly tested to come up with the desired output. The simulation feature was given emphasis during the tests. All errors encountered were debugged and improved in the integration testing of the simulator. After testing, the final stage was the publication of the C++ Code Structure and Algorithm Simulator.

4.1.1 Description of the Simulator

The study C++ Code Structure and Algorithm Simulator was a computer-aided instructional material for the students and instructors of the subject Introduction to Computer Science of Bicol University Polangui Campus. It aimed to provide supplementary instructional material for the enhancement of the teaching methodologies of the instructors in the aforementioned programming subject. The contents of the simulator were based from the approved course syllabus of the campus, course description and outline provided by the Commission on Higher Education.

The simulator was a flash application that simulates the coding structure of the C++ language. It includes exercises which the user will manually code based on certain requirements. This application immersed students in the different concepts of the C++ Programming Language. The features of the simulator include concepts, examples and three programming problems categorized as easy, average and difficult for each topic of the course.

The simulator was a computer program that simulates the coding structure of the C++ language. It includes exercises which the user will manually code based on certain requirements. This application immersed students in the different concepts of the C++ Programming Language. The features of the simulator include concepts, examples and three programming problems categorized as easy, average and difficult for each topic of the course.

4.1.2 Structure of the Simulator
The simulator had two major components: content discussions and programming problems for simulation. The introduction or splash screen of the simulator gave an overview of the description and features of the study. With this feature, the user has the option to proceed or exit the application. Figure 1 shows the splash screen of the simulator.

Figure 1. Splash screen / prelude of the simulator

Figure 2 shows the main screen of the simulator. It includes 10 major buttons that allow the users to access different modules of the system. These are the Instructions, About C++, cout-cin, if-else, switch, do-while, while, for loop, array and credits. Further, exit button found at the lower left corner offers a way to quit the program.

Figure 2. Home/main screen

Upon loading the main screen, instructions and reminders on how to use the simulator is preloaded. In addition, the user can still go back on this content by clicking the Instructions button.

Introductory lessons on the brief history and structure of C++ programming language can be viewed on the About C++ module shown in Figure 3. The lessons like on how to declare variable and usage of different operators like arithmetic, logical and relational were discussed in this module. Further, the examples on each operator were also included for detailed understanding of the concepts of the cited topics.

Figure 3. Content of cin-cout module

Discussions were included in each major topic of the simulator: cin-cout, if-else, switch, for, while, do-while, and array statement. Figure 3 provides picture of how a topic were presented in the simulator (cin-cout). The descriptions, syntax and usage of the topics were included. In addition, it includes programming examples and sample output for better understanding.

Additional buttons for programming problems were present in each of the focus of the simulator. The programming problems were categorized as easy, average and difficult. Easy problems are those that need very minimal manipulation and very little C++ structures. Then, average programming problems are more difficult than easy problems by having a more complicated algorithm. Lastly, difficult problems are those with the most complicated problems and use a lot of C++ structure.

Shown in Figure 4 is an example of a simulated problem. This module provides the programming problem or situation, sample output and working area for entering the answer in each situational statement.
Figure 4. Problem simulation home screen

Figure 5 shows the instruction (comments) that the user must satisfy. The numbers with “undone” are those lines that need to be answered based on the statement at the right side of the screen.

Figure 5. Instructions base on comments on the right side of the screen

When the user entered the correct syntax or structure for each line, the “Undone” word will be changed to “Done” (see Figure 6). On the other hand, if the user entered a statement with a compile error, an error message will appear at the lower left corner of the screen as shown in Figure 7. The user will not be able to proceed to the next instruction to be executed unless he successfully solved a problem.

Figure 6. Done will appear after a correct inputted C++ code

When the user successfully completed all the lines to be accomplished, an output window for simulation will appear (see Figure 8). This feature provides the user to have an actual experience on what to expect after solving a program. Moreover, the window offers choices for the user to go back to the main window or to proceed to the next programming problem of the active topic or content.

When the user successfully finished a programming problem, the button for that particular activity will be disabled. This feature enables the user to try more programming problems of the same topic but of different level or activities under the topic.

Figure 7. Error Message

After all the three programming problems has been solved in a topic or module. The button of the topic will change to DONE (see Figure 9).

Figure 8. Simulated output

Figure 9. Done with the topic

4.2 Functionality of the Simulator
The researcher performed several tests to determine the functionality of the simulator. It was tested again by five instructors to achieve unbiased result. The result showed that the simulator is 100% functional.

4.3 Acceptability of the Simulator

Table 2 shows the summary of the evaluation made by the respondents. Each item was interpreted verbally according to the value of the arithmetic mean obtained. The results showed that the simulator is “strongly acceptable” for both the students and instructors which have average weighted means of 4.65 and 4.57 respectively. The overall weighted mean of 4.61 showed the combine results of the respondents, which was interpreted as “strongly acceptable.”

4.4 Significant difference in the level of acceptability of the simulator

The t-test was used to determine the significant difference in the level of acceptability of the students and instructors. The result showed a computed value of 1.064 which is lower than the table of critical t-values of 0.05 which is 2.042. This means that the null hypothesis was accepted. Hence, there is no significant difference between the level of acceptability of the students and instructors.

5. CONCLUSIONS

The following conclusions were taken from the findings of the study:
1. The developed simulator contains discussions of the concepts, sample programs and three simulated problems. These inclusions were based from the course syllabus of the subject Introduction to Computer Science and guided by the Commission on Higher Education minimum requirements.
2. The simulator was functional.
3. The simulator was strongly acceptable for both students and instructors as an instructional material.
4. The null hypothesis (H₀) was accepted and there is no significant difference between the acceptability of the students and instructors. Hence, the simulator, and the content and programming problems were comprehensible on the level of both students and instructors.

6. RECOMMENDATIONS

Based from the summary, findings and conclusions of the study, the following recommendations were proposed:

1. Additional examples with illustrative output and programming problems may be added. Also, saving of students’ activity and editing of the contents be possible. Further, topics on advanced programming may be added to cover the topics for Computer Programming 2.
2. Further debugging especially on error handling of the simulator is recommended.
3. Study on the performance of the students using this instructional material is highly recommended.
4. Study on the significant difference between the mean score of the pre-test and post-test of a CAI and non-CAI.
Table 2. Test of acceptability of the simulator

<table>
<thead>
<tr>
<th>Item</th>
<th>STUDENTS</th>
<th></th>
<th></th>
<th>INSTRUCTORS</th>
<th></th>
<th></th>
<th>OVERALL</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Verbal</td>
<td>Mean</td>
<td>Verbal</td>
<td>Mean</td>
<td>Verbal</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Interpretation</td>
<td>Interpretation</td>
<td>Interpretation</td>
<td>Interpretation</td>
<td>Interpretation</td>
<td>Interpretation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Part I. Simulator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Easy to use</td>
<td>4.80</td>
<td>Strongly</td>
<td>4.60</td>
<td>Strongly</td>
<td>4.70</td>
<td>Strongly</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Contributes to the students’ learning process</td>
<td>4.80</td>
<td>Strongly</td>
<td>5.00</td>
<td>Strongly</td>
<td>4.90</td>
<td>Strongly</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Improves the students’ level of understanding of the concepts</td>
<td>4.47</td>
<td>Strongly</td>
<td>4.80</td>
<td>Strongly</td>
<td>4.63</td>
<td>Strongly</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Part II. Content and Programming Problems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. The topic is reflected in the course syllabus</td>
<td>4.77</td>
<td>Strongly</td>
<td>4.80</td>
<td>Strongly</td>
<td>4.78</td>
<td>Strongly</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. The terms used match the students’ level of understanding</td>
<td>4.60</td>
<td>Strongly</td>
<td>4.80</td>
<td>Strongly</td>
<td>4.70</td>
<td>Strongly</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. The discussion of the topic is:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a. Clear</td>
<td>4.60</td>
<td>Strongly</td>
<td>4.40</td>
<td>Strongly</td>
<td>4.50</td>
<td>Strongly</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b. Specific</td>
<td>4.50</td>
<td>Strongly</td>
<td>4.60</td>
<td>Strongly</td>
<td>4.65</td>
<td>Strongly</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c. Comprehensible</td>
<td>4.40</td>
<td>Strongly</td>
<td>4.40</td>
<td>Strongly</td>
<td>4.50</td>
<td>Strongly</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. The procedure is:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a. Brief</td>
<td>4.63</td>
<td>Strongly</td>
<td>4.40</td>
<td>Strongly</td>
<td>4.52</td>
<td>Strongly</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b. Simple</td>
<td>4.63</td>
<td>Strongly</td>
<td>4.60</td>
<td>Strongly</td>
<td>4.62</td>
<td>Strongly</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c. Easy to interpret</td>
<td>4.57</td>
<td>Strongly</td>
<td>4.40</td>
<td>Strongly</td>
<td>4.48</td>
<td>Strongly</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>II. Objectives are:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Clear</td>
<td>4.77</td>
<td>Strongly</td>
<td>4.20</td>
<td>Acceptable</td>
<td>4.48</td>
<td>Strongly</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Specific</td>
<td>4.73</td>
<td>Strongly</td>
<td>4.20</td>
<td>Acceptable</td>
<td>4.47</td>
<td>Strongly</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Attainable</td>
<td>4.47</td>
<td>Strongly</td>
<td>4.20</td>
<td>Acceptable</td>
<td>4.33</td>
<td>Strongly</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average Weighted Mean</td>
<td>4.65</td>
<td>Strongly</td>
<td>4.57</td>
<td>Strongly</td>
<td>4.61</td>
<td>Strongly</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td>Acceptable</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

REFERENCES


